Throwing Down The Gauntlet: A Discussion Of Techniques For Bounding Advanced Tracking Algorithm Performance

John E. Gray
Naval Surface Warfare Center
Dahlgren Division
Dahlgren, VA, U.S.A.

A. Sunshine Smith-Carroll
Naval Surface Warfare Center
Dahlgren Division
Dahlgren, VA, U.S.A.

Abstract - ***For many applications of radar and sensor based filtering, simulations can not represent the sole estimate of performance, provide points where threats become engageable, or determine when to use weapons' platform based sensors effectively in an engagement, etc... No significant advances have been proposed to analytically characterize performance or at least bound performance of the Kalman filter other than the use of simple two or three state constant gain filters. This paper suggests methods for characterizing filter algorithms that can be used to bound the advanced tracking algorithms that are used in a single sensor or muli-sensor environment.***

Keywords: Target Tracking, Estimation, Fusion, Constant Gain Filters.

1 Introduction

Significant advances have been made in the last forty years in target tracking that have enhanced radar based tracking well beyond Kalman's publication of the two papers that define Kalman filtering. At the same time that significant advancements techniques have been proposed by theoreticians to deal with the real problems of target tracking that the Kalman filter has proven inadequate to solve. However, no significant advances have been proposed to analytically characterize filter performance or at least bound performance of these filters other than the usage of simple two and three state constant gain filters. This is a problem for many applications of radar and sensor based filtering, because system performance characterization cannot depend on simulations to estimate performance on the fly, so to speak, or to provide points when threats become engageagble, or to determine when to use weapon's platform based sensors effectively in an engagement, etc.

The cost of a statistically significant sampling the statistical universe in which a weapons-sensor platforms exist is prohibitive due to the expanse of simulation time required to achieve adequate sampling which has significant financial cost. Thus, there is always a need for analytical approaches to characterize the interaction between filters and the sensor operational environment. The reasons for insistence on analytical methods over pure simulation are many, but it is sufficient to note that without useful rules of thumb, apprentice engineers produce filter design that are clearly nonsense to those practitioners who have gained experience over many years with functioning filters that are correctly designed. These rules of thumb become even more important in the multi-sensor fusion environment, where the level of experience and history of performance of the algorithms is limited. The general characteristics that bound the performance of the fusion algorithms are small compared to the statistical universe that the algorithm will be operating in.

Thus, it would be valuable to be able to characterize tracking algorithm performance in the single sensor and multi-sensor environment so that a useful characterization of performance is available. This provides the ability of a user to perform sanity checks of filter outputs for incorporation in the control loop of the sensor-weapons platform and examine weapon effectiveness. This paper suggests some methods for characterizing filter algorithms that can be used to bound the advanced tracking algorithms that are used in a both a single sensor multi-sensor environments. In addition, this paper attempts to throw down the gauntlet to other practitioners to present methods of their own on some future occasion.

2 Constant Gain Filters

The tracking equations for the $\alpha - \beta$ filter (the necessary background is found in the books by Bar-Shalom [2] and Blackman [4]) consist of two parts: prediction equations, which are given by

$$X_p(k) = AX_s(k-1) \tag{1}$$

where

$$X_p(k) = \begin{bmatrix} x_p(k) \\ v_p(k) \end{bmatrix} \tag{2}$$

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

and smoothing equations, which are given

$$X_s(k) = FX_s(k-1) + G\,x_m(k) \tag{4}$$

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Throwing Down the Gauntlet: A Discussion of Techniques for Bounding Advanced Tracking Algorithm Performance	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center,Dahlgren Division,Dahlgren ,VA,22448**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
9th International Conference on Information Fusion, 10-13 July 2006, Florence, Italy. Sponsored by the International Society of Information Fusion (ISIF), Aerospace & Electronic Systems Society (AES), IEEE, ONR, ONR Global, Selex - Sistemi Integrati, Finmeccanica, BAE Systems, TNO, AFOSR's European Office of Aerospace Research and Development, and the NATO Undersea Research Centre.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

where

$$X_s(k) = \begin{bmatrix} x_s(k) \\ v_s(k) \end{bmatrix}, \tag{5}$$

$$F = \begin{bmatrix} 1-\alpha & (1-\alpha)T \\ -\frac{\beta}{T} & 1-\beta \end{bmatrix}, \tag{6}$$

and

$$G = \begin{bmatrix} \alpha \\ \frac{\beta}{T} \end{bmatrix}. \tag{7}$$

- $x_s(k)$ = smoothed position at the k-th interval
- $x_p(k)$ = predicted position at the k-th interval
- $x_m(k)$ = measured position at the k-th interval
- $v_s(k)$ = smoothed velocity at the k-th interval
- $v_p(k)$ = predicted velocity at the k-th interval
- T = radar update interval or period
- α, β = filter weighing coefficients

The question of the selection of filter coefficient values and the relationship between the coefficients used by tracking filters to determine pointing commands for a tracking radar dates back at least as far as work by Sklansky [16]. Sklansky proposed performance measures including stability, transient response, noise and maneuver error as a function of the dynamic parameters α and β. All of the work was based on a frequency domain or z-transform analysis. Benedict-Bordner[3] proposed a relationship between α and β based on a pole-matching technique that combined transient performance and noise reduction capability. Analysis performed by Simpson [15], Neal, and Benedict [14] extended this analysis to the $\alpha-\beta-\gamma$ filter. Later, much of this work was summarized in the open literature by Kalata [11]. A summary of subsequent developments in the literature to 1992 is found in Kalata [11] with some additional work since then found in Gray [7] , and in the open literature.

There are several different dynamics models that lead to an $\alpha - \beta$ filters with different statistical and performance attributes. While all filters have the same noise reduction ratios for position and velocity, they have different transient responses or bias depending on which threat model one uses. In matrix form, the predicted update is (note this model lumps maneuverability uncertainty in to the velocity component)

$$X(k+1) = \Phi X(k) + \Psi \tilde{w}(k), \tag{8}$$

while the measurement model in matrix form is

$$z(k+1) = HX(k+1) + \widetilde{n}(k+1); \tag{9}$$

where

$$\Psi = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{10}$$

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \tag{11}$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{12}$$

$$X(k) = \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}, \tag{13}$$

which is termed **Model 1** for the form of Ψ. Another approach to modeling maneuver uncertainty is to incorporate it into both the position and velocity prediction components: in matrix form (**Model 2**)

$$X(k+1) = \Phi X(k) + \Psi_2 \widetilde{w}_a(k) \tag{14}$$

where

$$\Psi_2 = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \tag{15}$$

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \tag{16}$$

$$X(k) = \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}, \tag{17}$$

and H is the same as before. In [8], a method is developed for solving the Lyapunov equations for these type of models. For the steady state performance, the noise reduction ratios are the position (K_x), velocity (K_v), and position-velocity cross term (K_{xv}) which are given as

$$K_x = \frac{2\alpha^2 + \beta(2-3\alpha)}{D}, \tag{18}$$

$$K_v = \frac{2\beta^2}{DT^2}, \tag{19}$$

$$K_{xv} = \frac{\beta(2\alpha-\beta)}{DT}, \tag{20}$$

where $D = \alpha(4-2\alpha-\beta)$. For **Model 2**, the noise reduction ratio for the position and the noise reduction ratio for the velocity are given as before while the transient reduction ratios are

$$T_x^2(0) = \frac{(1-\alpha)^2 T^4}{2\alpha\beta} \tag{21}$$

and

$$T_v^2(0) = \frac{T^2}{4}\left[\frac{2\alpha^2 - 3\alpha\beta + 2\beta}{\alpha\beta}\right]. \tag{22}$$

Note, this method applies to any constant gain filter, so higher state filters can be characterized as well by solving the Lyapunov equation analytically. Thus, noise covariance, both steady state noise reduction and transient response have analytical models which characterize performance by this method. The various transient noise reductions are useful for estimating filter performance when un-modeled dynamic behavior occurs. When combined with the noise reduction ratios, these provide performance envelopes that bound filter performance. In addition to bounding performance, the Jacobian of the noise reduction and transient performance ratios allow one to determine one of four possible relationship (Benedict-Bordner, Kalata [11], Continuous White Noise, and an unnamed one) between the filter coefficients, e.g. $\alpha = \alpha(\beta)$ as discussed in [9].

The other aspect of filter design is selection of the filter gains. This can be accomplished by introducing a cost function methodology. A function of mean squared error is taken to be the arbitrator of performance. While mean square is widely accepted, other

normed distance measure could be used. A mean squared cost function can be formed by a combination of noise reduction and filter response to an un-modeled term [9]. A normalized cost function (J_N) (normalized means the mean squared error equals the normalized cost times the normalization constant N which for the steady state velocity cost function is $\left(\frac{\sigma^2}{T^2}\right)$). In general, all cost functions are of the form

$$J_N(\alpha,\beta) = f(\alpha,\beta) + g(\alpha,\beta)\Lambda^2, \tag{23}$$

where f, g are functions of α and β that depend on which **Model 1** is being considered and Λ is (a_0 is the un-modeled acceleration and σ is the measurement noise)

$$\Lambda^2 = \frac{a_0^2 T^4}{\sigma^2}. \tag{24}$$

where a_0 is the un-modeled acceleration. A normalized cost function based on velocity noise reduction ratio and the **Model 2** velocity transient noise reduction ratio is

$$J_N^V(2,\alpha,\beta) = \frac{2\beta^2}{\alpha(4-2\alpha-\beta)} + \lambda_2^V(\alpha,\beta)\Lambda^2 \tag{25}$$

where

$$\lambda_2^V(\alpha,\beta) = \frac{(1-\alpha)^2}{2\alpha\beta}. \tag{26}$$

A cost function based on position noise reduction ratio and the **Model 2** position transient noise reduction ratio is

$$J_N^P(2,\alpha,\beta) = \frac{2\alpha^2-\beta(3\alpha-2)}{\alpha(4-2\alpha-\beta)} + \lambda_2^P(\alpha,\beta)\Lambda^2, \tag{27}$$

where

$$\lambda_2^P(\alpha,\beta) = \frac{2\alpha^2-3\alpha\beta+2\beta}{\alpha\beta}. \tag{28}$$

Another combination for a possible cost functions based on the steady state lags can be achieved by combining the position noise reduction ratio and position lag which gives,

$$J_N^P(S,\alpha,\beta) = \frac{2\alpha^2-\beta(3\alpha-2)}{\alpha(4-2\alpha-\beta)} + l_x^2\Lambda^2, \tag{29}$$

where

$$l_x = \frac{1-\alpha}{\beta}; \tag{30}$$

while the steady state normalized velocity lag cost function is given by:

$$J_N^V(S,\alpha,\beta) = \frac{2\beta^2}{\alpha(4-2\alpha-\beta)} + l_v^2\Lambda^2, \tag{31}$$

where

$$l_v = \frac{\alpha}{\beta} - \frac{1}{2}. \tag{32}$$

The acceleration bias of the smoothed velocity response of the filter or bias is given by:

$$L_v = a_0 T l_v = a_0 T\tau \tag{33}$$

where the velocity lag coefficient is defined as τ. The acceleration bias of the smoothed position response filter is

$$L_x = a_0 T^2 l_x, \tag{34}$$

where the position lag coefficient is defined as

$$l_x = \frac{1-\alpha}{\beta} \equiv \frac{1}{\beta} - \tau - \frac{1}{2}. \tag{35}$$

Note both β and α can always be written in terms of lag coefficients as

$$\beta = \frac{2}{2(l_x+\tau)+1}, \tag{36}$$

and

$$\alpha = \frac{(2\tau+1)}{2(l_x+\tau)+1}, \tag{37}$$

so $\frac{\alpha}{\beta} - \frac{1}{2} = \tau$ which does not depend on which relationship one chooses between α and β. The noise reduction ratios expressed in the lag form can be expressed as

$$P_v(l_x,\tau) = \frac{2}{(2l_x+\tau)(2\tau+1)}, \tag{38}$$

and

$$P_x(l_x,\tau) = \frac{(2\tau^2+2l_x+\tau)}{(2l_x+\tau)(2\tau+1)} \tag{39}$$

For the **Model 2** transient terms we have

$$\lambda_2^P(\alpha,\beta) = \frac{2\left(2\tau^2+2l_x+\tau\right)}{(2\tau+1)} \tag{40}$$

and

$$\lambda_2^V(\alpha,\beta) = \frac{l_x^2}{(2\tau+1)}. \tag{41}$$

Examining the filter coefficient relationships in terms of the velocity lag coefficient τ, gives several interesting results. For each of the filter coefficient relationships,we can express the filter coefficients in terms of the lag τ coefficient form (e.g. the form that depends just on τ and not l_x). The velocity lag is the same for all three coefficient relationships, but $l_x(\tau)$ is different. Also, only for the Kalata relationship does $l_x(\tau) \underset{\tau\to 0}{\to} 0$, the others don't. Also if we define $\tau' = T\tau$, then if we replace τ with $\frac{\tau}{T}$ and then replace T with an arbitrary update interval T_k and still have [13]

$$\alpha_k = 1 - \frac{\tau^2}{(T_k+\tau)}, \tag{42}$$

and

$$\beta_k = \frac{2{T_k}^2}{(T_k+\tau)}. \tag{43}$$

Thus the Kalata filter coefficient relationship is preserved for arbitrary aperiodic updates T_k, which maintains the nominal filter gains and preserves the filter coefficient relationship. The lag characteristics suggest a difference between the properties of the filter coefficient relationships based on position lag, which suggests that the Kalata relationship is preferred in our multi-platform application because of communication and computation delays that are mitigated for the

Kalata relationship only. Once a particular filter coefficient relationship is chosen $\alpha = \alpha(\beta)$, the cost function can be expressed in terms of τ. The cost function can then be minimized to give $\tau = \tau(\Lambda)$ and hence $\alpha = \alpha(\Lambda)$.

Thus filter performance can be expressed in a tractable form based on one design parameter Λ. Since all of this can be accomplished using a minimum mean square approach, a very good characterization of a single filter boundaries exists based on an analytical characterization, provided a good estimate of Λ exists. Thus, the ability to match any single model Kalman filter performance to within a few percent exists (Dale Blair—private communication). Early work on extending these concepts to three state filters are found in [15] and [14]; but the methods presented here illustrated by the $\alpha - \beta$ filter are easily extended to any multiple-state constant gain filter. Furthermore, the label constant gain for the filter is a bit miss-leading. Since $\alpha = \alpha(\Lambda)$, and we can always update an estimate of Λ as frequently as necessary up to including the sensor update rate to maintain desired track accuracy. So designing a filter this way provides a tight bound of performance relative to the single stage filter one is trying to arrive at a bound performance for. For multiple model filters, the situation becomes more complicated.

For a multiple model filter, such as the Interacting Multiple Models (IMM), the analytical approach is also helpful. Many tracking simulations are run using the IMM with two models: constant velocity and nearly constant velocity, or nearly constant velocity and constant acceleration. It is difficult to decide which transition matrix is optimal, but it is clear that certain choices are definitely sub-optimal. Matrices near those suggested by the sojourn time calculation behave well. However, while perturbing the matrix a small amount away from the suggested values does result in a change in performance, it is difficult to decide whether the change is for the better, since performance is not measured by one number, but by a balance of competing interests. The results obtained imply that having three models is not necessary except in unusual cases. Also, the values of the off-diagonal elements do not matter, except for the requirement that the rows sum to one. Recall the IMM algorithm consists of six steps:

1. **Weights update:**

$$\mu_{i|j}(k-1|k-1) = \frac{1}{c_j} p_{ij} \mu_i(k-1) \tag{44}$$

This is a calculation of the probability that model i was in effect at time $k-1$, given that model j is in effect at time k. In the above equation, the value c_j is a normalizing constant, p_{ij} is the probability of a transition from model i to model j, and $\mu_i(k-1)$ is the probability that model i was in effect at time $k-1$.

2. **Mix input estimates and covariance matrices:**

$$\widehat{x}^{0j}(k-1|k-1) = \sum_i \widehat{x}^i(k-1|k-1)\mu_{i|j}(k-1|k-1) \tag{45}$$

$$\begin{aligned} P^{0j}(k-1|k-1) = \\ \sum_i \mu_{i|j}(k-1|k-1) \cdot \{P^i(k-1|k-1) + \\ \left[\widehat{x}^i(k-1|k-1) - \widehat{x}^{0j}(k-1|k-1)\right] \times \\ \left[\widehat{x}^i(k-1|k-1) - \widehat{x}^{0j}(k-1|k-1)\right]' \} \end{aligned}$$

Each filter produces an estimate of position, and if the filter were operating alone, that estimate would be used during the next iteration as an initial value. In the IMM algorithm, these estimates are mixed together, so that the input to filter j is the estimate of the position most likely at time $k-1$, given that model j is in effect at time k.

3. **Apply filters:**

Now the estimates and covariance matrices calculated in the step before, and the observation taken at time k, are used as input to the Kalman filters. Each filter behaves normally at this point.

4. **Compute model likelihood functions:**

$$\Lambda_j(k) = P(z(k)|M_j(k), Z^{k-1}) \tag{46}$$

The likelihood function will be used to update the probabilities of the various models. The likelihood of model j at time k is defined to be the probability of observing the value that was actually observed, given the previous history and the assumption that model j was in effect at time k.

5. **Compute model probabilities:**

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) \sum_i p_{ij} \mu_i(k-1) \tag{47}$$

This calculation represents using the observation at time k to update of the probability of each model being in effect. The value c is a normalizing constant.

6. **Compute updated state estimate and covariance matrix:**

$$\widehat{x}(k|k) = \sum_i \widehat{x}^i(k|k)\mu_i(k), \tag{48}$$

$$\begin{aligned} P(k|k) \quad = \quad & \sum_i \mu_i(k|k)\{P^i(k|k) + \left[\widehat{x}^i(k|k) - \widehat{x}(k|k)\right] \times \\ & \left[\widehat{x}^i(k|k) - \widehat{x}(k|k)\right]'\}. \end{aligned} \tag{49}$$

This calculation is only necessary if an output of a composite estimate and covariance matrix is desired. These values are not used elsewhere in the algorithm.

The Kalman filter steps can be replaced with two different filter models using different analytical models based on either transient or steady state performance. For a three model case, one would use a three state analytical filter. Noting that for the IMM algorithm, the matrix $[p_{ij}]$ calculates the model probabilities for the final output. The existence of this matrix stems from a representation of the target's maneuvers as transitions from state to state in a Markov chain. Thus, it makes sense to apply Markov theory in the analysis of this problem. In a Markov chain, τ_i is the sojourn time, that is, the expected amount of time continuously spent in state i, and is given by

$$\tau_i = \frac{1}{1-p_{ii}}. \tag{50}$$

This follows by noting [5]:

$$E(\text{Time in } i| \text{ State at time 0 is } i) = \frac{1}{1-p_{ii}}. \tag{51}$$

Thus, the IMM filter performance can be viewed as a weighted combination of individual analytical filters with the weights based on the sojourn time of a particular filter. While not tested yet, it is expected that this should yield fairly accurate estimate of IMM performance boundaries and will be the subject of a future publication.

3 Fusion Algorithms

In order to improve the quality of the state estimate, local tracks or data can be communicated from the sensors' platform to a central site for the purpose of estimation fusion. The results of this paper apply to an arbitrary communication rate. The sensor platforms can communicate their data/track after every update, after a given number of updates, or after every time period. Communication delays exist between the sensor platforms and the fusion center. Let $t = t_{k-1}$ be the last time the fusion center performed a track fusion and let $t_k = t_{k-1} + T_f$ be the time of the next fusion time. The period T_f is an adaptive design parameter. It can be as small as the time it takes to receive two tracks from two different sensor platforms or as long as the time it takes to receive as many tracks as the total number of sensors in the network. The value of T_f mainly depends on the priority given to the track fusion task which depend on the application at hand. The number n of data/tracks to be fused during the interval $[t_{k-1},\ t_k]$ is an arbitrary number between 2 and m, where m is the number of senors used. Because of the difference in communication delays, some of the validated tracks may arrive out-of- sequence. To deal with the out-of- sequence tracks without special processing, all the tracks that arrive at the fusion center during the time $[t_{k-1}, t_k]$ after removing redundant tracks, are fused simultaneously. *The asynchronous multi-sensor track fusion problem can be stated as follows:*

Given a number of asynchronous valid tracks, $\{X_i(t_{k_i}|t_{k_i}),P_i(t_{k_i}|t_{k_i})\}$, $i = 1, 2, ..., n$, *that arrive at the fusion center during the time interval* $[t_{k-1}, t_k]$, *find the best estimate in the minimum mean square sense of the system state at time* t_k *when it is computed according to the fusion rule.*

$$X_f(t_k|t_k) = \sum_{i=1}^{n} L_i(t_{k_i}) X_i(t_{k_i}|t_{k_i}) \tag{52}$$

where the L_i*'s are unknown weighting matrices to be determined.* The error of the fused track at time $t_k = kT_f$ is

$$\tilde{X}_f(t_k|t_k) = X_f(t_k|t_k) - X(t_k) \tag{53}$$

Using the system dynamics and the fusion rule gives

$$\begin{aligned}\tilde{X}_f(t_k|t_k) \quad = \quad & \sum_{i=1}^{n} L_i \tilde{X}_i(t_{k_i}|t_{k_i}) + \left[\sum_{i=1}^{n}[L_i\Phi(t_{k_i},t_k) - I\ ]\right] \times \\ & X(t_k) - \sum_{i=1}^{n} L_i \Phi(t_{k_i},t_k) W_{t_{k_i}}^{t_k}\end{aligned} \tag{54}$$

If all the local filters are unbiased, then

$$\left[\sum_{i=1}^{n}[L_i\Phi(t_{k_i},t_k) - I\ ]\right] = 0, \tag{55}$$

so the fused track is unbiased if

$$\sum_{i=1}^{n} L_i \Phi(t_{k_i},t_k) = I \tag{56}$$

This represents the first constraint on the choice of the weighting matrices. Therefore,

$$L_n = \Phi(t_k,t_{k_n}) - \sum_{i=1}^{n-1} L_i \Phi(t_{k_i},t_{k_n}). \tag{57}$$

The error covariance matrix of the fused track can be defined as

$$P_f(t_k|t_k) = E\{\tilde{X}_f(t_k|t_k)\tilde{X}_f(t_k|t_k)^{'}\} \tag{58}$$

which allows determination of the weights L_i which define the optimal asynchronous track fusion filter. Further details are found in [1]

Theorem 1: The error covariance matrix of the fused track using the fusion rule is given by

$$P_f(k|k) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} L_i M_{ij} L_j' + \sum_{i=1}^{n-1} L_i N_i + \sum_{i=1}^{n-1} N_i L_i^{'} + M_n \tag{59}$$

Proof: *See* [1].

Theorem2 : The minimum mean square solution of the asynchronous track fusion problem using the fusion rule is given by

$$X_f(t_k|t_k) = \sum_{i=1}^{n} L_i X_i(t_{k_i}|t_{k_i}) \tag{60}$$

$$P_f(t_k|t_k) = LML^{'} + LN + N^{'}L^{'} + M_n \quad (61)$$

where

$$[L_1\ L_2\ ...\ L_{n-1}] = -N^{'}M^{-1} \quad (62)$$

$$L_n = \Phi(t_k, t_{k_n}) - \sum_{i=1}^{n-1} L_i \Phi(t_{k_i}, t_{k_n}). \quad (63)$$

Proof: *See* [1].

The reason for this example is that it is indicative of all fusion algorithms, they are variations on a similar theme. With the proper redefinitions of matrices, both track and data fusion amount to the same thing: **weighted (positive semi-definite and the sum is normalized to one) combinations of data**. The results can be extremely complicated, difficult to understand, difficult to predict in terms of performance, and difficult to determine the algorithm's underlying correctness, but they have the same underlying mathematical form when understood properly. Given this observation, an alternative suggests itself as a means of understanding these types algorithms based on maximum entropy analysis to interpret the probabilities as the solution to a maximum entropy problem. There are two approaches to developing this methodology which will now be review.

4 Maximum Entropy Procedure As A Means Of Understanding Fusion Schemes

The degree of uncertainty (which is equivalent to the surprise value) in the information is defined as the entropy:

$$S[|p\rangle\,; n] == -k\,\langle p|\ln p\rangle = -k\sum_{i=0}^{n} p_i \ln p_i; \quad (64)$$

Most applications do not permit one to measure the probabilities, p_i, associated with a physical variable, $f(x_i)$, instead the expected value, $\langle f(x)\rangle$, is measured. Probabilities are connected to expected values by using the formula for $\langle f(x)\rangle$. The goal of a maximum entropy procedure is to find an assignment to the probability vector $|p\rangle$ subject to the following two conditions: **Condition 1:** $|p\rangle$ is normalized: $\langle 1|p\rangle = 1$.**Condition 2:** Known assignment of the means ($r = 1, 2, 3...m$): $\langle p|g_r\left(|x\rangle\right)\rangle = \langle g_r(|x\rangle)\rangle = \alpha_r$.

These two conditions provide two equations, so $(n-2)$ more are needed. One approach is to find an assignment of probabilities that maximizes the entropy. These two condition plus the principle [10]: "*The distribution $|p\rangle$ that maximizes the uncertainty in the expected value subject to the constraint of the available information*" provides such an assignment. The method of Lagrange undetermined multipliers is a standard method for solving an optimization problem subject to constraints. The function one wants to maximize is the entropy, so it is adjoined to the free parameters (Lagrange undetermined multipliers) times the equations for the constraints to form a Lagrangian, which for entropy maximization is (take $k = 1$)

$$\begin{aligned} L &= -\langle p|\ln p\rangle - \lambda_0\left(\langle 1|p\rangle - 1\right) \\ &\quad -\sum_{r=1}^{m}\lambda_r\left[\langle p|g_r\left(|x\rangle\right)\rangle - \alpha_r\right]. \end{aligned} \quad (65)$$

To minimize with respect to the probabilities, one computes the particular state $\langle p|e_i\rangle$ such that $\frac{\partial L}{\partial\langle p|e_i\rangle}$is a minimum:

$$\frac{\partial L}{\partial\,\langle p|e_i\rangle} = 0 = \ln p_i + 1 + \lambda_0 + \sum_{r=1}^{m}\lambda_r g_r\left(x_i\right). \quad (66)$$

Solving for the state's probability $\langle p|e_i\rangle = p_i$ gives the probability assignment as

$$\begin{aligned} \langle p|e_i\rangle &= e^{-\left(\lambda_0+1+\sum_{r=1}^{m}\lambda_r g_r(x_i)\right)} \\ &= e^{\left[-(\lambda_0+1+\langle\lambda|g(x_i)\rangle)\right]}. \end{aligned} \quad (67)$$

Substituting probabilities into Condition 1 gives first Lagrange undetermined multiplier as

$$\begin{aligned} e^{(\lambda_0+1)} &= \sum_{i=1}^{n} e^{\left(-\sum_{r=1}^{m}\lambda_r g_r(x_i)\right)} \\ &= \sum_{i=1}^{n} e^{-\langle\lambda|g(x_i)\rangle}. \end{aligned} \quad (68)$$

Other Lagrange undetermined multipliers are obtained by substituting probability assignment into Condition 2 in the Appendix,

$$\langle p|g_r\left(|x\rangle\right)\rangle = \sum_{i=1}^{n} e^{-\langle\lambda|g(x_i)\rangle} g_r(x_i) = \alpha_r e^{(\lambda_0+1)}\ , \quad (69)$$

which solves the problem of assigning the probabilities. Define the partition function as Z as

$$e^{(\lambda_0+1)} = Z, \quad (70)$$

The Hamiltonian H is defined as

$$H_i = \sum_{r=1}^{m}\lambda_r g_r\left(x_i\right), \quad (71)$$

$$= \langle\lambda|g(x_i)\rangle\,. \quad (72)$$

The probability assignment becomes

$$\langle p|e_i\rangle = p_i = \frac{\exp(-H_i)}{Z}, \quad (73)$$

which is a familiar expression to physicists.

A general model for any process that can be viewed as a weighted graph (communication or fusion network for example), e.g. the connected graph has a weight w_{ij} associated with each arc (i, j), we can associate a number that can be defined as a probability, or as a fusion rule for different sources or as a multiple model

filter with the data drawn from different sources, by the assignment :

$$P_{ij} = \begin{cases} \frac{w_{ij}}{\sum_j w_{ij}} & \text{if } (i,j) \text{ is an arc} \\ 0 & \text{if } (i,j) \text{ is not an arc} \end{cases}. \quad (74)$$

So instead of interpreting $\lambda_i x_i(G)$ as the Lagrange undetermined multiplier times some property of the graph, it can be interpreted as interaction potential between components i and j. The λ_i is interpreted as scaling parameter μ, a field coupling parameter, or the inverse temperature. It is possible to posit a variety of interaction models, work out their consequences, and work out the equivalent probability distributions. The simplest form for the Hamiltonian is when the expected number of edges $\langle m \rangle$ is known, so $H(G)$ becomes

$$H_\mu(G) = \mu m(G). \quad (75)$$

This model is trivial, so next consider the simplest model where the adjacency matrix $A(G)$ has components a_{ij}

$$a_{ij} = \begin{cases} 1 & \text{if } i \text{ is connected to } j \\ 0 & \text{if } i \text{ is not connected to } j \end{cases}. \quad (76)$$

The number of edges m is

$$m(G) = \sum_{i<j} a_{ij}(G), \quad (77)$$

therefore the Hamiltonian is given by:

$$H(G) = \mu \sum_{i<j} a_{ij}(G). \quad (78)$$

The partition function (PF) is

$$\begin{aligned} Z_\mu(G) &= \sum_G \exp(-H(G)) \\ &= (1+\exp(-\mu))^{\binom{n}{2}}. \end{aligned} \quad (79)$$

In general it is possible to replace μ_i with μ_{ij}, so the Hamiltonian is $H(G) = \sum_{i<j} \mu_{ij} a_{ij}(G)$ while the Partition function is

$$Z_{\vec{\mu}}(G) = \prod_{i<j} \left(1 + e^{-\mu_{ij}}\right). \quad (80)$$

The free energy, which is the logarithm of the partition function, is

$$\begin{aligned} F_\mu &= -\ln(Z_\mu(G)) \\ &= -\sum_{i<j} \ln\left(1+e^{-\mu_{ij}}\right). \end{aligned} \quad (81)$$

The first moment is

$$\langle f \rangle = \frac{dF_\mu}{d\mu} = \frac{1}{(1+e^{\mu_{ij}})}, \quad (82)$$

while the standard deviation is

$$\sigma_f^2 = \frac{d^2 F_\mu}{d\mu^2} = -\frac{e^{\mu_{ij}}}{(1+e^{\mu_{ij}})^2}. \quad (83)$$

In general, can specify interaction Hamiltonian as

$$H = \sum_{i<j} \Lambda_{ij} \sigma_{ij} \quad (84)$$

where Λ_{ij} is parameter that couples each edge together and the degree sequence satisfies $\nu_i = \sum_j \sigma_{ij}$. The partition function is

$$Z = \prod_{i<j} \left(1 + e^{-\Lambda_{ij}}\right). \quad (85)$$

Free energy is

$$F = -\sum_{i<j} \ln\left(1+e^{-\Lambda_{ij}}\right). \quad (86)$$

Note probability of an edge between edge i and j is the expected number of the degree sequence, then

$$p_{ij} = \langle \sigma_{ij} \rangle = \frac{\partial F}{\partial \Lambda_{ij}} = \frac{1}{(1+e^{\Lambda_{ij}})}. \quad (87)$$

Underlying basis for the parameterization that gives us Bernoulli model with

$$P(G) = p_{ij}^m (1-p_{ij})^{\binom{n}{2}-m}. \quad (88)$$

A term $\sum_i C_i \theta_i$ to the Hamiltonian without changing the probability model for the graph while allowing us to relate $\langle C_i \rangle$ to the characteristic of the graph interaction. Thus, this work can be extended to social networks [6],[17], [18], by broadening it to include network modeling of interest. *Any interaction model that can be cast into the above form is a hidden Markov model.* We also note that any weighting scheme used in a fusion algorithm for a tracking filter can be cast in the form of an interaction network model, so

1. In principle, any probability assignment model can be thought of as a physical interaction model, so one can bring to bear the full power of statistical physics on it.

2. The translation between physics interaction Hamiltonian to graph theory probability models needs to be made more transparent.

3. This approach applies to any type of network that can be modeled as a graph with a normalized or zero/one entries for the adjacency matrix.

Some areas for further development of the interaction model method include:

1. While this field has rich predictive capability, the question is usefulness when applied to specific problem domains.

2. Time evolving graphs have the potential to be a huge application area with many different types of domain applications.

3. This technique can be applied to flow networks by using a non-symmetric matrix.

4. An exponential probability model is equivalent to Markov Model for time evolution. Deviations from the exponential model can be thought of as "memory" in the system under consideration, so the question is what role memory plays in the system interaction model.

In creating a network that fuses weighted information from different sources we are creating the equivalent of a physical interaction system that has an underlying physics that we can strive to understand and exploit. Thus a fusion algorithm when looked at this way is physics model based on interpreting our weighing of the data from different sources as an underlying interaction model.

5 Conclusions

Any assignment of probabilities can always be viewed as the solution to a Bayesian Maximum Entropy problem. Given that is the case, then the question can be asked: "What is the Maximum Entropy problem that an assignment of fusion weights is the answer to?" For other examples not related to this question, see the suggestive paper by Kesavan [12] Being able answer this question for various fusion algorithms, would allow us classify them in the same fashion. This would lead to an ordering of algorithms in terms of the complexity of the problem they solve. This will be dealt with in a subsequent paper. Our ideal has been to suggest new means to understand the strengths and limitations shared information based on tracking data provided from sensor networks. The goal is to find better means for sharing resources, so information can be allocated across all or part of the platforms within the network based on a maximum entropy viewpoint.

References

[1] A. T. Alouani, J. E. Gray, and D. H. McCabe, "Theory of Distributed Estimation Using Multiple Asynchronous Sensors", T-AES Vol. 41, No. 2, April 2005.

[2] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association,* Academic Press, Inc., Orlando, FL (1988).

[3] T. Benedict and G. Bordner, "Synthesis of an Optimal Set of Radar Track-While Scan Smoothing Equations," *IRE Trans. on Automatic Control*, AC-7, pp. 27-32, 1962.

[4] S. S. Blackman, *Multiple Target Tracking with Radar Applications*, Artech House, 1986.

[5] L. Bloomer and J. E. Gray, "Are More Models Better?: The Effect of the Model Transition Matrix on the IMM filter", Thirty-Fourth Southeastern Symposium on System Theory - March 18, 2002.

[6] Frank, O. and Strauss, D., "Markov Graphs", *Journal of the American Statistical Association*, 1Vol. 81, 832-842, 1986.

[7] J. E. Gray and G. J. Foster, "Solutions of Matrix Equations Arising in Track Filter Theory," *Proceedings of The Thirtieth Southeastern Symposium on System Theory,* University of West Virginia in Morgantown, 8-10 Mar, 1998.

[8] J. E. Gray, S. Smith-Carroll, L. A. Jordan, "The Solution to the Lyapunov Equation in Constant Gain Filtering and Some of its Applications", *Proceedings of The Thirty-Sixth Southeastern Symposium on System Theory,* Georgia Technical University, Atlanta Georgia, 20-23 March, 2004.

[9] J. E. Gray, W. J Murray, and S. Smith-Carroll, "What do filter coefficient relationships mean?", *Proceedings of The Thirty-Sixth Southeastern Symposium on System Theory,* Georgia Technical University, Atlanta Georgia, 20-23 March, 2004.

[10] E. T. Jaynes, in *Maximum Entropy and Bayesian Methods*, edited by C.R. Smith, G. J. Erickson, and P.O. Neudorfer, Kluwer Academic, 1992.

[11] P. Kalata, "$\alpha - \beta$ Target Tracking Systems: A Survey," *ACC Proceedings*, 1992, pp. 832-836.

[12] H. K. Kesavan and J. N. Kapur, "The Generalized Maximum Entropy Principle", IEEE Trans. on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept/Oct, 1989.

[13] W. J. Murray and J. E. Gray, "Target Tracking with Explicit Control Filter Lag", " in Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory, March 9-11, 1997, Cookeville, Tennessee, pp. 81-85.

[14] S.R. Neal and T.R. Benedict, "Parametric Relations for the $\alpha - \beta - \gamma$ Filter Predictor," *IEEE Trans. Auto. Cont.*, AC-12, pp. 315-317, Jun 1967.

[15] H.R. Simpson, "Performance Measures and Optimization Conditions for a Third-Order Sampled-Data Tracker," *IEEE Trans. Auto. Cont.*, AC-8, pp. 182-183, Apr 1963.

[16] J. Sklansky, "Optimizing the performance of a track-while-scan system," *RCA Rev.*, pp. 163-185, Jun 1957.Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association,* Academic Press, Inc., Orlando, FL (1988).

[17] Skvoretz, J., "Complexity Theory and Models for Social Networks", *Complexity*, Vol. 8, No. 1, 2003.

[18] Strauss, D., "On a General Class of Models for Interaction", *SIAM Review*, Vol. 28, No. 4, 1986.